**ENRICHED COACHING CENTER**

**A PROJECT REPORT Submitted**

**partial fulfilment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**by**

**Niharika Maheshwari**

**(University Roll No : 1900290149066 )**

**Under the Supervision of**

### Ms. Neelam Rawat

**Associate Professor**



**to the**

**Department of Computer Applications**

**Dr. A.P.J ABDUL KALAM TECHNICAL UNIVERSITY**

**LUCKNOW**

**July, 2021**

**CERTIFICATE**

Certified that **Niharika Maheshwari** (**1900290149066**) has carried out the project work presented in this project report entitled “**Coaching Center**” for the award of Master of Computer Application from Dr. A.P.J. Abdul Kalam Technical University, Lucknow under my supervision. The report embodies result of original work, and studies are carried out by the student himself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University.

**Ms. Neelam Rawat External Examiner**

Associate Professor

Dept. of Computer Applications

KIET Group of Institutions, Ghaziabad

**Dr. Ajay Kumar Srivastava**

Professor & Head

Department of Computer Application

KIET Group of Institutions, Ghaziabad

Date:

**ENRICHED COACHING CENTER**

**ABSTRACT**

**Coaching centres** offer **classes** as per subjects and help students to prepare well. **Coaching centres** offering specialized **classes** for Joint and Management entrances are already very popular. In the last few years many students have cracked entrance exams successfully after studying at these **coaching centres**.

**Coaching** is the skill of providing ongoing and specific feedback in a supportive manner for employee learning, development and improvement. A process aimed at exerting a positive influence to improve the behavior or performance of a person.

Coaching classes are always played a role in proper guidance. If you want to stay in the competition then you need a regular and routine life without having any stress. Don’t force your children to attend the classes. Make them understand that these types of classes have  Actually attended a coaching class has lots of benefits which will teach you more and make you update more as this type of classes is full with fresh flocks of students and their ideas and curiosity.

**ACKNOWLEDGEMENT**

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Ms. Neelam** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Ajay Kumar Srivastava, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**Niharika Maheshwari**

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Certificate | | ii |
|  | Abstract | | iii |
|  | Acknowledgements | | iv |
|  | Table of Contents | | v-vi |
|  | List of Tables | | vii |
|  | List of Figures | | viii |
|  | **CHAPTER 1: INTRODUCTION** | | **9-17** |
|  |  | 1.1 Project Description | 9-10 |
|  |  | 1.2 Project Scope | 10 |
|  |  | 1.3 Hardware/Software used in Project | 11 |
|  |  | 1.3.1 Technology Description | 12-17 |
|  | **CHAPTER 2: LITERATURE REVIEW** | | **18-23** |
|  |  | 2.1 Proposed System | 20 |
|  |  | 2.2 Technical Feasibility | 21 |
|  |  | 2.3 Operational Feasibility  2.4 Economic Feasibility | 22  22 |
|  |  | 2.5 Behavioural Feasibility | 23 |
|  |  |  |  |
|  |  |  |  |
|  | **CHAPTER 3: DATABASE MANAGEMENT** | | **24-26** |
|  | 3.1 Schema Builder  3.2 Flow Graphs | | 25  26 |
|  | **CHAPTER 4: OBJECT INTERFACES** | | **27-41** |
|  | 4.1 Interface (Screenshot) | | 27-41 |
|  | **CHAPTER 5: Extend Salesforce with Clicks, Not Code** | | **42-46** |
|  | 5.1 App Setup | | 42-46 |
|  | **CHAPTER 6: TESTING**  6.1 Salesforce Exploratory Testing  6.2 Salesforce Test Automation  6.3 Salesforce Load Testing 6.4 Salesforce Security Testing | | 47-53  51  52  52  53 |
|  | **CHAPTER 7: REFERENCES** | | **54** |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Name of Table** | **Page** |
| 1.1 | Hardware | 11 |
| 1.2 | Software | 11 |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Figure No.** | **Name of Figure** | **Page No.** | |
| Fig 3.1 | Schema Builder | | 25 | |
| Fig 3.2.1 | Zero level DFD | | 26 | |
| Fig 3.2.2 | One level DFD | | 26 | |
| Fig 4.1 | Contact Object | | 27 | |
| Fig 4.1.1 | Adding New Contact | | 28 | |
| Fig 4.1.2 | Contact Detail | | 29 | |
| Fig 4.2 | Report and Dashboard Object | | 30 | |
| Fig 4.2.1  Fig 4.3 | Create New Report  Select Coaching Questionnaire Record Type | | 31  32 | |
| Fig 4.4 | Process Builder | | 33 | |
| Fig 4.5.1  Fig 4.5.2  Fig 4.6.1  Fig 4.6.2  Fig 5.1 | Classic Email Templates  Classic Email Templates  Classic Letterheads  Classic Letterheads  App Setup | | 34-36  37-38  39-40  40-41  43 | |
| Fig 5.2 | App Setup | | 44 | |
| Fig 5.3 | App Setup | | 45 | |
| Fig 5.4 | App Setup | | 46 | |

**CHAPTER 1**

**INTRODUCTION**

* 1. **Project Description**

The new way of building and running applications are enabled by the world of cloud computing, where we can access applications, over the Internet as utilities, rather than as pieces of software running on your desktop or in the server room. This model is already quite common for consumer apps like email and photo sharing, and for certain business applications, like customer relationship management (CRM).The Force.com platform is the world's first Platform as a Service (PaaS), enabling developers to create and deliver any kind of business application in the cloud, entirely on-demand and without software. It's a breakthrough new concept that is making companies radically more successful by letting them translate their ideas into deployed applications in record time. Building, sharing, and running business applications have never been so easy. Building and running business applications with traditional software has always been too complex, slow, and expensive. A new model, called cloud computing, has emerged over the last decade to address this problem. Applications that run in the cloud are delivered as a service so companies no longer have to buy and maintain hardware and software to run them. Salesforce.com pioneered this model with applications business over the last decade. More recently, Force.com have opened up the infrastructure and made it available for anyone building any business application and running it on the servers using the Force Platform. The Force Platform allows you to store structured data, implement business logic with workflow rules, approval processes and custom code, support Web browsers, integrate with other applications, do reporting and analytics and scale up or down—all with sub second response time, high availability, and security you need to run your mission critical business apps. Data are store on cloud where all the information will be stored safely and access anywhere, anytime by authorize person.

* 1. **Project Scope**

This is an application project for understanding and demonstrating cloud computing capabilities using Force.com. I use Sales Force features to achieve the goal.The objective of this application is to develop and deploy an application in the sales force environment.

This application project must satisfy all type of users and should enable all cloud computing capabilities. It is assumed that this application will be developed in the following environment:

* Force.com environment
* Visualforce as the developing language
  1. **Hardware / Software used in the Project**

**Table 1.1 Hardware**

|  |  |
| --- | --- |
| **Hardware** | **Configuration** |
| Processor | Intel® Core™ i3-7100U CPU @2.4GHz |
| RAM | 4GB |
| Monitor | HP |
| Modem | Internet Connectivity |

**Table 1.2 Software**

|  |  |
| --- | --- |
| **Software** | **Configuration** |
| Operating System | Windows XP /7/8/10 |
| Software | Chrome, Microsoft Edge |

* + 1. **Technology Description**

**SALESFORCE**

**Salesforce.com, Inc.** is an American cloud-based software company headquartered in [San Francisco, California](https://en.wikipedia.org/wiki/San_Francisco,_California). It provides [customer-relationship management](https://en.wikipedia.org/wiki/Customer-relationship_management) (CRM) service and also sells a complementary suite of enterprise applications focused on customer service, marketing automation, analytics, and application development.

Salesforce is the primary enterprise offering within the Salesforce platform. It provides companies with an interface for case management and task management, and a system for automatically routing and escalating important events. The Salesforce customer portal provides customers the ability to track their own cases, includes a social networking [plug-in](https://en.wikipedia.org/wiki/Plug-in_(computing)) that enables the user to join the conversation about their company on [social networking Web sites](https://en.wikipedia.org/wiki/Social_networking_service), provides analytical tools and other services including email alert, Google search, and access to customers' entitlement and contracts

#### **Service Nature of Salesforce:**

1. **Salesforce as SaaS (Software as a Service):** There is no need for installation, setup or downloads but you just have to log in and start using the software apps across the cloud. Isn’t it more convenient and amazing? The answer is a big “YES” when using Salesforce CRM for your business.
2. **Salesforce as PaaS (Platform as a service):** Here, there is no need for a separate platform, but you can use code created by other developers to deploy apps. Obviously, you have to customize it as per your business needs, but it saves a lot of time and money as demanded by businesses today.
3. **Salesforce as IaaS (Infrastructure as a service):** Here, there is no need for installing any hardware or software program, but data and apps are stored securely on the cloud. Even you don’t have to take the backup, but the cloud will take care of it automatically.

**1.3.1.1 APEX**

Apex is a proprietary programming language provided by the Force.com platform to developers similar to [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) and [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)). It is a strongly typed, object-oriented, case-insensitive programming language, following a dot-notation and curly-brackets syntax. Apex can be used to execute programmed functions during most processes on the Force.com platform including custom buttons and links, event handlers on record insertion, update, or deletion, via scheduling, or via the custom controllers of Visualforce pages.

Due to the multitenant nature of the platform, the language has strictly imposed governor limitations to guard against any code monopolizing shared resources. Salesforce provides a series of [asynchronous processing](https://en.wikipedia.org/wiki/Asynchrony_(computer_programming)) methods for Apex to allow developers to produce longer-running and more complex Apex code.

**1.3.1.2 LIGHTNING**

Salesforce made public the [front end](https://en.wikipedia.org/wiki/Front_and_back_ends) of its platform, called Lightning. This component-based framework is what the Salesforce mobile app is built on. Salesforce built on this framework in 2015 by releasing the Lightning Design System, an HTML style framework with default CSS styling built in. This framework allows customers to build their own components to either use in their internal instances or sell on the AppExchange.

The Salesforce Lightning App Builder is a tool for [rapid application development](https://en.wikipedia.org/wiki/Rapid_application_development) of responsive web interfaces. This interface allows for different screens to be put together based on Lightning components. This can be used as layouts for records or specific applications.

Lightning Experience, released in 2016, is the new redesigned interface in Salesforce for processes enhancement. Since then all the apps available on AppExchange need to be Lightning and those built on Classic have to migrate to Lightning as Classic is not to be updated any more by Salesforce. The platform offers an option for developers to employ migration techniques to enable the new user interface and switch to Lightning.

**1.3.1.3 JavaScript**

**JavaScript** (often shortened to **JS**) is a lightweight, interpreted, object-oriented language with first-class functions, and is best known as the scripting language for Web pages, but it's used in many non-browser environments as well. It is a prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles.

JavaScript runs on the client side of the web, which can be used to design / program how the web pages behave on the occurrence of an event. JavaScript is an easy to learn and also powerful scripting language, widely used for controlling web page behavior.

Contrary to popular misconception, **JavaScript is not "Interpreted Java"**. In a nutshell, JavaScript is a dynamic scripting language supporting prototype based object construction. The basic syntax is intentionally similar to both Java and C++ to reduce the number of new concepts required to learn the language. Language constructs, such as if statements, for and while loops, and switch and try ... catch blocks function the same as in these languages (or nearly so).

JavaScript can function as both a procedural and an object oriented language. Objects are created programmatically in JavaScript, by attaching methods and properties to otherwise empty objects **at run time**, as opposed to the syntactic class definitions common in compiled languages like C++ and Java. Once an object has been constructed it can be used as a blueprint (or prototype) for creating similar objects.

JavaScript's dynamic capabilities include runtime object construction, variable parameter lists, function variables, dynamic script creation (via eval), object introspection (via for ... in), and source code recovery (JavaScript programs can decompile function bodies back into their source text).

**1.3.1.4. SOQL**

SOQL is Salesforce Object Query Language that is highly similar to SQL (Structured Query Language). With the help of SOQL, you can always search the organizational data wisely. It can be combined with APEX, Visualforce, or Force.com IDE too. While writing a query using SOQL, you should use SELECT command and a lot of other conditions too.

The SOQL can be used when you have the basic idea of objects and related data. It can be used to retrieve data from one or multiple objects that are connected together based on requirements. It helps in getting the total count and sort queries as well. Further, it can be combined with SOSL (Salesforce Object Search Language) to search the organizational data when they are not sure about objects.

SOSL is programmed in such a way to work on the basis of the search index. To improve the performance of searching, you should combine SOQL and SOSL together because searching work is done better by SOSL as compared to the SOQL.

**CHAPTER 2**

**LITERATURE REVIEW**

As users of the Internet, we're all familiar with the fascinating, innovative, creative, and sometimes silly ways in which it has changed how we work and play. From social networking sites to wikis to blogs, and more, it’s exciting to watch the innovations taking place that are changing the ways we communicate and collaborate.

While these changes have certainly impacted how we work with content, a similar set of Internet-driven ideas and technologies is changing how we build and work with business applications.

While yesterday's business applications required thousands, if not millions, of dollars and sometimes years of professional services help to set up and customize, the technologies offered by the Internet today make it much easier to create, configure, and use business applications of all kinds. Indeed, the power of the Internet has given us the ability to solve new kinds of business problems that, because of complexity or cost, had previously remained out of reach.

Just as the changes that moved publishing technology from paper to bits made it possible for us to have information about anything in the whole world right at our fingertips, the changes in application technology make it similarly possible to imagine a robust, enterprise-class application for almost any business need. Sound pretty good? Then you're probably wondering: “What's the magic that makes this possible?”

These new ways of building and running applications are enabled by the world of cloud computing, where you access applications, or apps, over the Internet as utilities, rather than as pieces of software running on your desktop or in the server room.

This model is already quite common for consumer apps like email and photo sharing, and for certain business applications, like customer relationship management (CRM).

By using the existing system we have to maintain a high configuration system to run some huge software’s which is used to develop the applications. This may need huge cost to buy and maintain.

We have to buy the each and every software’s which are used for developing the applications. Also we have to update the software’s for latest version. This needs maintenance risks such as economical risk and technical risk.

The main disadvantage of the existing system over cloud environment is lack of portability. Even though we are having high configured systems and updated software’s we have to take the system with us when we are moving from one place to other place. This will create risk to the users.

The main disadvantage of the existing system is lack of disaster recovery. That means in the case of any disaster occurs all the data available in the system cannot be recovered. Thus it may leads to the loss of data.

The existing system also needs high initial out lay to set up the system.

**2.1. Proposed System**

Since there are lot of problems available with the existing system a new technology is used in this project called as **cloud computing.** The proposed system in this project is to develop an application in the cloud computing environment and have to deploy it and to launch the application in cloud.

The cloud environment used in this project is called as **salesforce environment**.

A simple application is developed in the cloud environment and deployed in the same environment using the tools provided there itself. The application is launched as a site inside the cloud. The security provided by the platform provider (salesforce) is used in this application.

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities.

**2.2. Technical Feasibility**

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

Does the existing technology sufficient for the suggested one?

Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So, there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

**2.3. Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following:

¬ Is there sufficient support for the management from the users?

¬Will the system be used and work properly if it is being developed and implemented?

¬Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So, there is no question of resistance from the users that can undermine the possible application benefits.

**2.4. Economic Feasibility**

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

The costs conduct a full system investigation.

The cost of the hardware and software.

The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

**2.5. Behavioral Feasibility**

An estimate should be made of how strong a reaction the user staff is likely to have towards the development of a computerized system. It is common knowledge that computer installation have something to do with Turnover, Transfers and changes in employee Job Status. Normal human psychology of human beings indicate that people are resistant to change and computers are known to facilitate change. Any project formulations should consider this factor also. Before the development of the Project titled "Delhi Metro", the need to study the feasibility of the successful execution of the project was felt and thus the following factors are considered for a Feasibility Study.

1. Need Analysis.
2. Provide the users information pertaining to the preceding requirement.

**CHAPTER 3**

**DATABASE MANAGEMENT**

A Database in Salesforce is defined as the organized collection of objects where each object contains some information. Data is stored in the form of database tables for people, things, contacts, etc that are important for any project in the future. Each database has a set of certain rows and columns where information is stored in the form of fields and records. It helps produce database systems

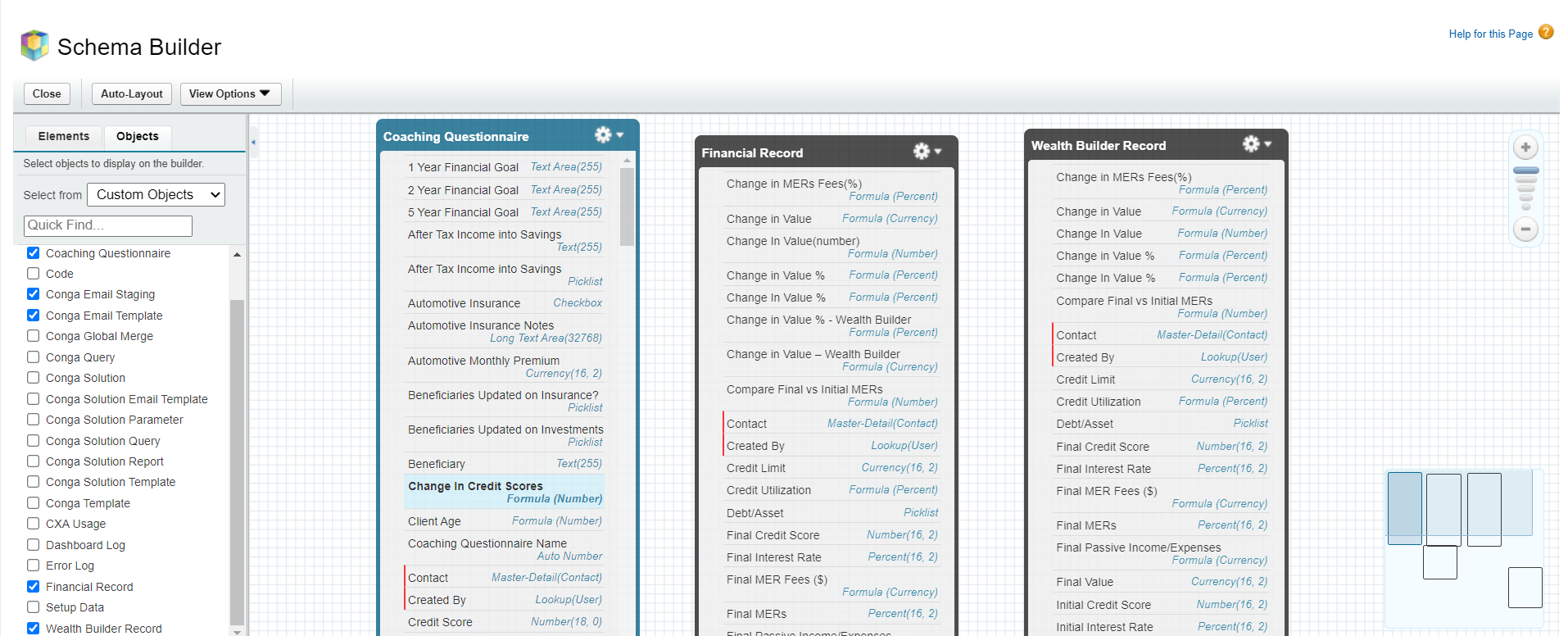
1. That meet the requirements of the users
2. Have high performance.

One such important concept in Oracle is the Salesforce database class. Database classes can specify either you want to continue the execution or not if some error is encountered. You can also add one Boolean parameter to the Salesforce database class to make it more functional. The same task can be completed using DML statements but you are not free to keep a check on query execution.

Another important concept is the Salesforce database architecture. It has the multi-tenancy architecture where clouds are used to share resources reliably and securely. The multitenant Salesforce database architecture has a huge impact on application delivery and the infrastructure. In simple words, we can say that Salesforce database architecture is highly similar to the architecture of relational database systems.

**3.1 Schema Builder**

Fig 3.1 Shows the relationship between Custom Objects.



**Fig. 3.1 Schema Builder**

**3.2. Flow Graphs**

**3.2.1 Zero Level DFD**

Select Questionnaire Type

Contact

Coaching Questionnaire

**Fig. 3.2.1 Zero Level DFD**

**3.2.2 One Level DFD**

Admin Portal

Coaching Questionnaire

Select Questionnaire Type

Fill Details

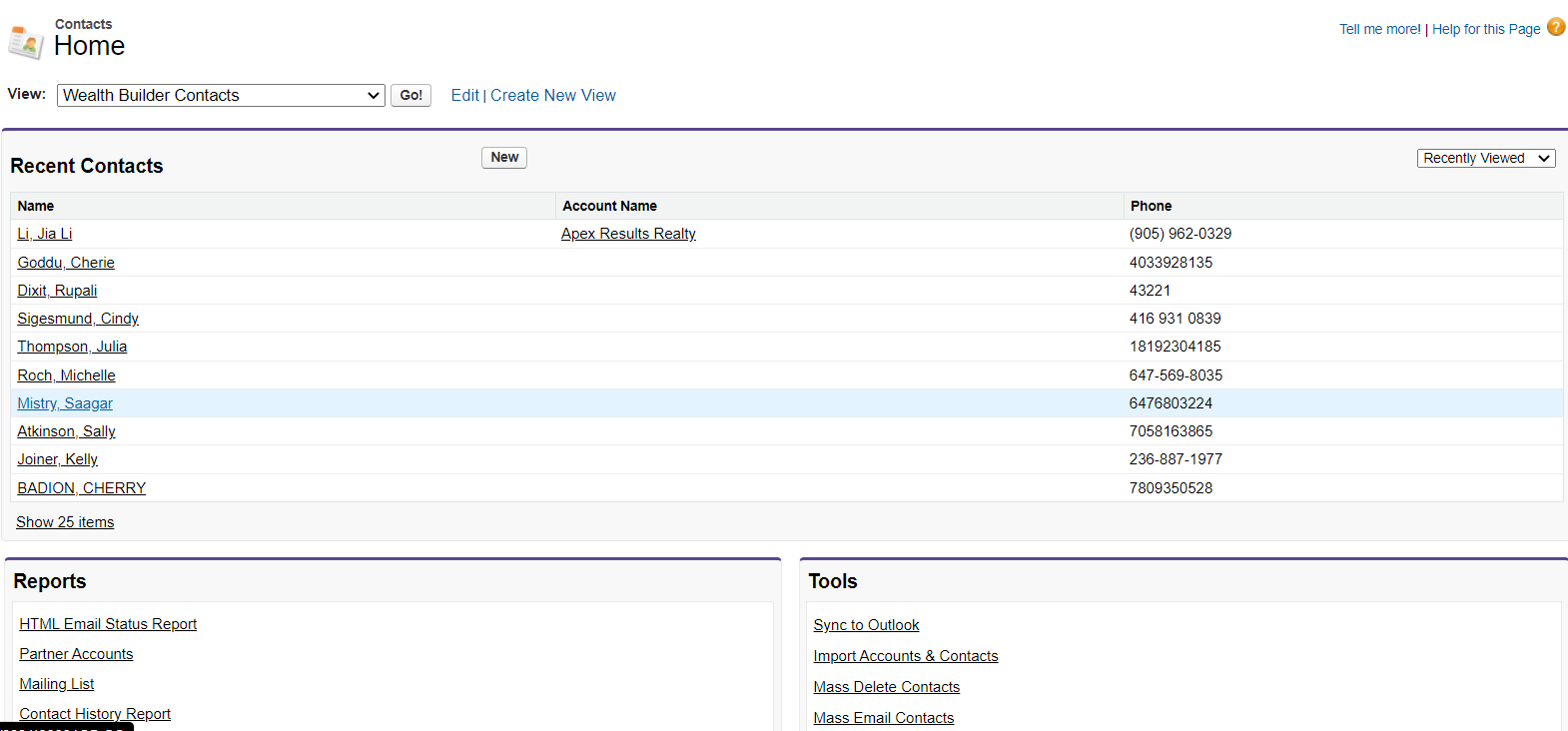
**Fig. 3.2.2 One Level DFD**

**CHAPTER 4**

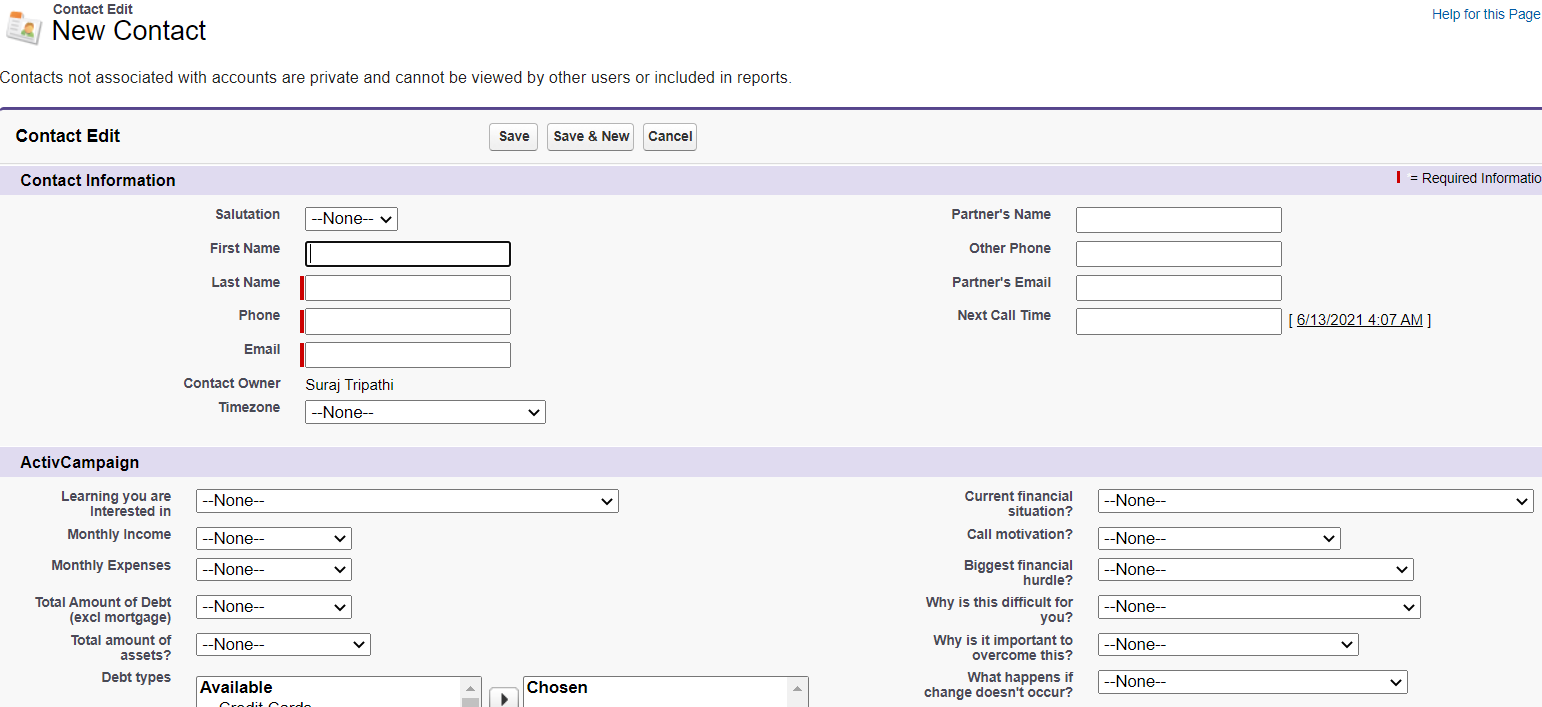
**OBJECT INTERFACES**

**4.1. Contact Object**

Fig. 4.1 It show all the recent contacts which can be added with this Coaching center.

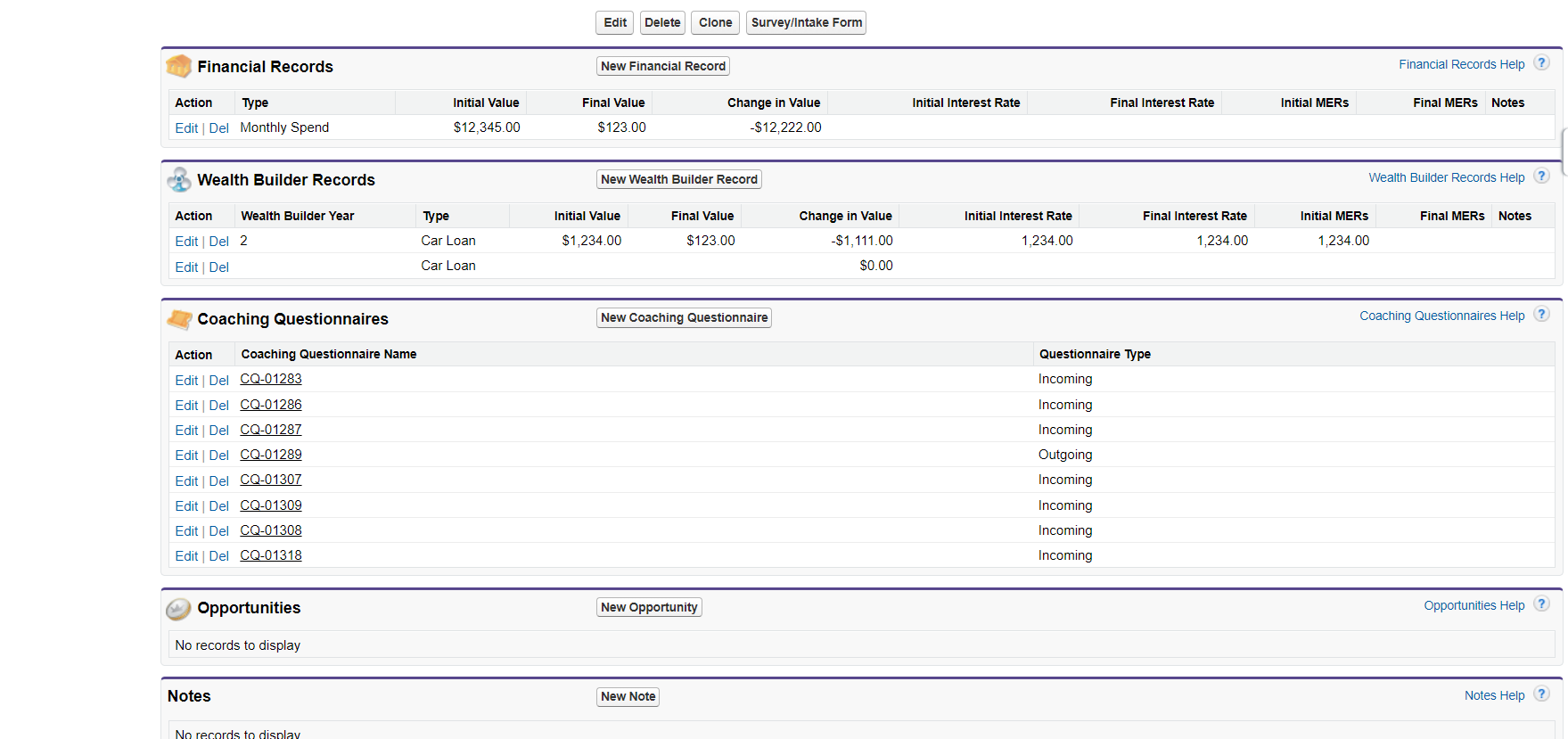


**Fig. 4.1 Contact Object Shows Recent Contacts**

Fig. 4.1.1. In this snapshot we can see to add Contact and along with coaching program details and other information. 

**Figure 4.1.1. Adding New Contact**

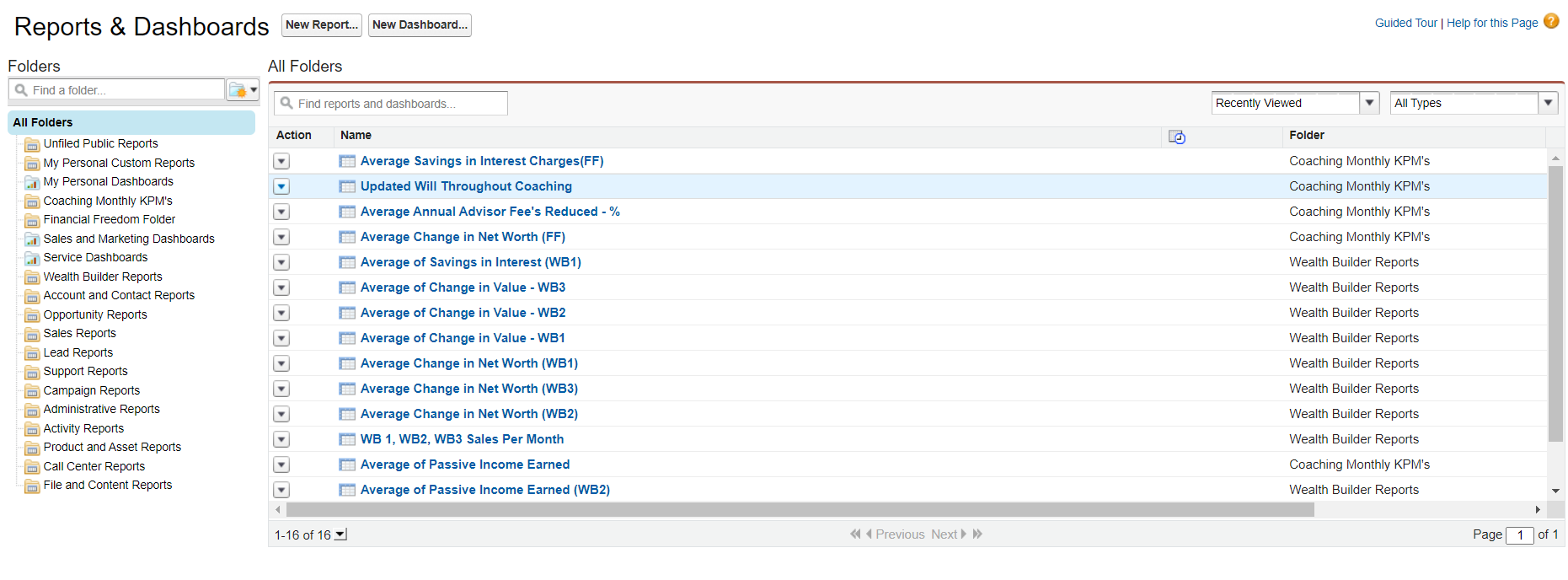
Fig. 4.1.2. This shows the financial records detail like type,initial value,final value,change in value etc. and show wealth builder records as well as coaching questionaires which contain questionnaire type like incoming and outgoing type.



**Fig. 4.1.2** **Contact Detail**

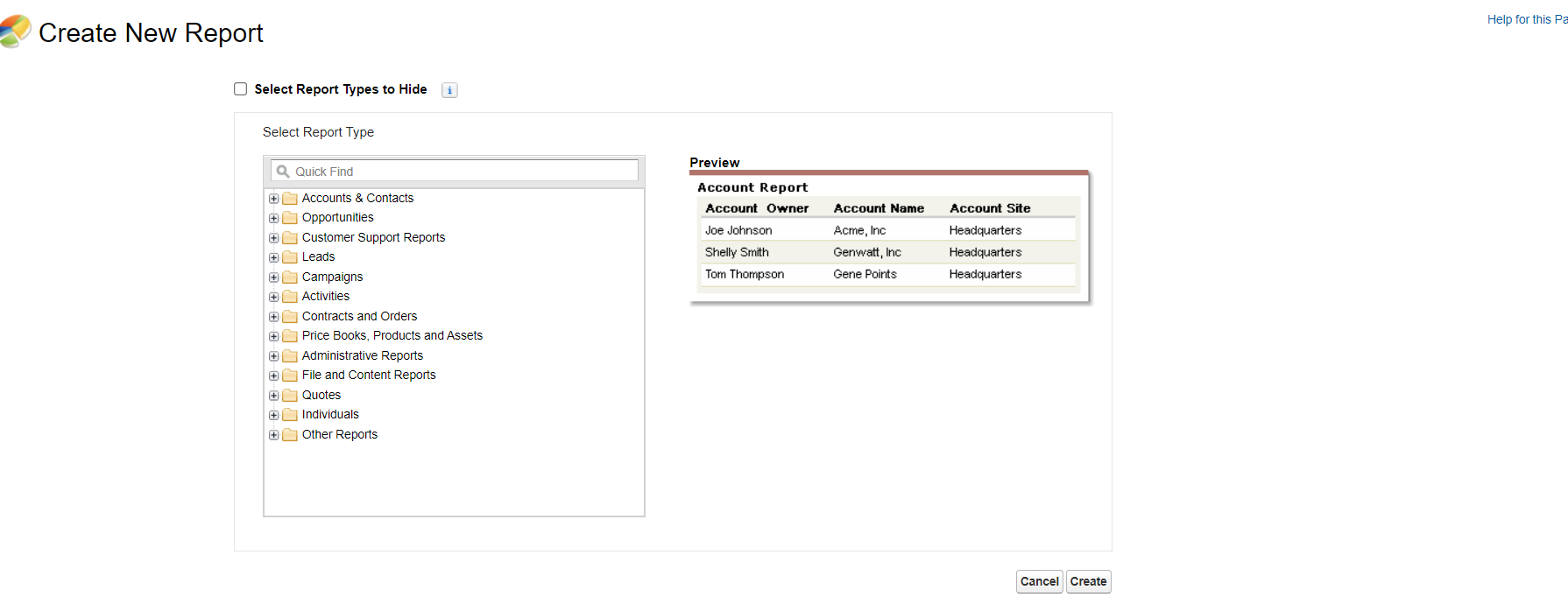
**Fig 4.2.** **Report and Dashboard object**

It shows all the reports and dashboards which are available in this coaching center which includes several folders which have been created.



**Fig. 4.2. Reports and Dashboard**

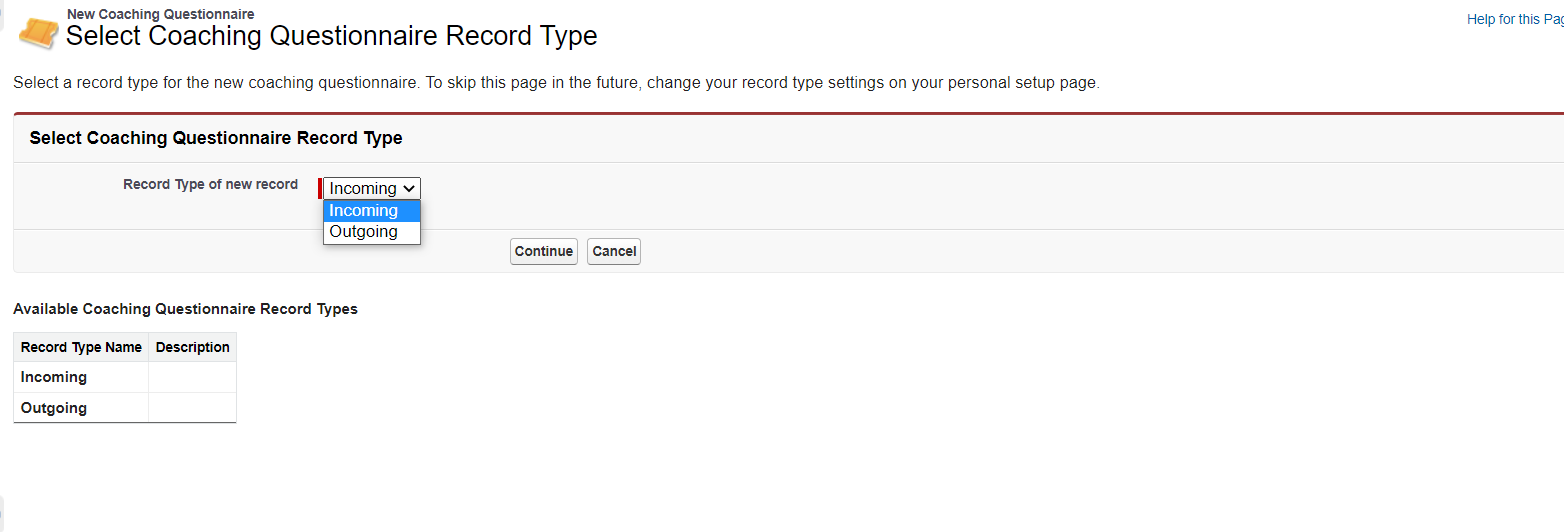
Fig 4.2.1. In this figure how we create new reports by selecting the report type like accounts and contacts , leads, opportunity, quotes etc and then select create option.



**Fig. 4.2.1 Create New Report**

**Fig. 4.3 SELECT COACHING QUESTIONNAIRE RECORD TYPE**

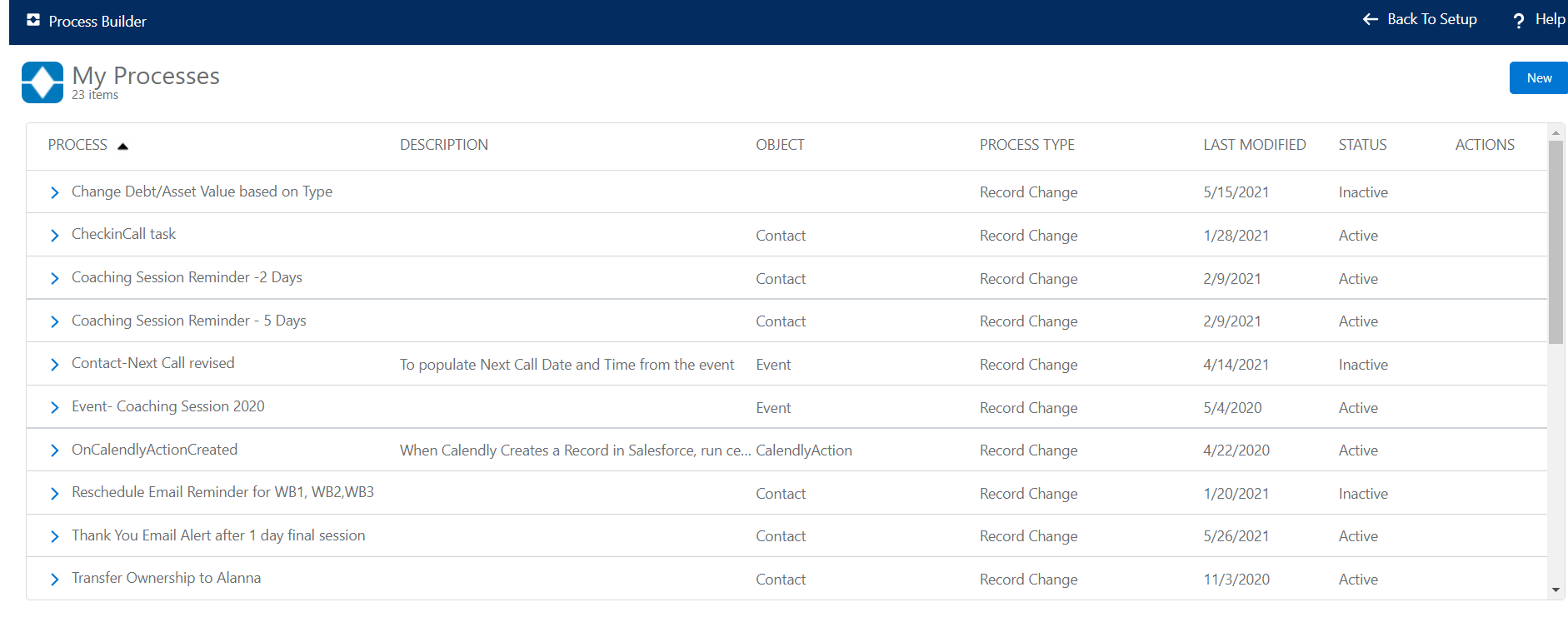
This object shows when a person selecting the coaching questionnaire it should be select which record type it fill the details like incoming and outgoing type and fill the details according to it.



**Fig. 4.3 Select Coaching Questionnaire Record Type**

**Fig. 4.4 PROCESS BUILDER**

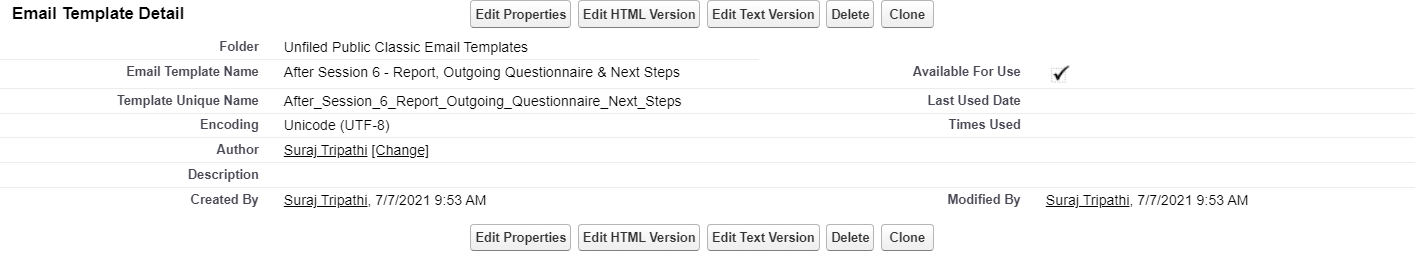
**Process builder** is used to automate more complex business **process** just by a few point and click actions. It provides a user – friendly representation for building a **process** in **Salesforce**.

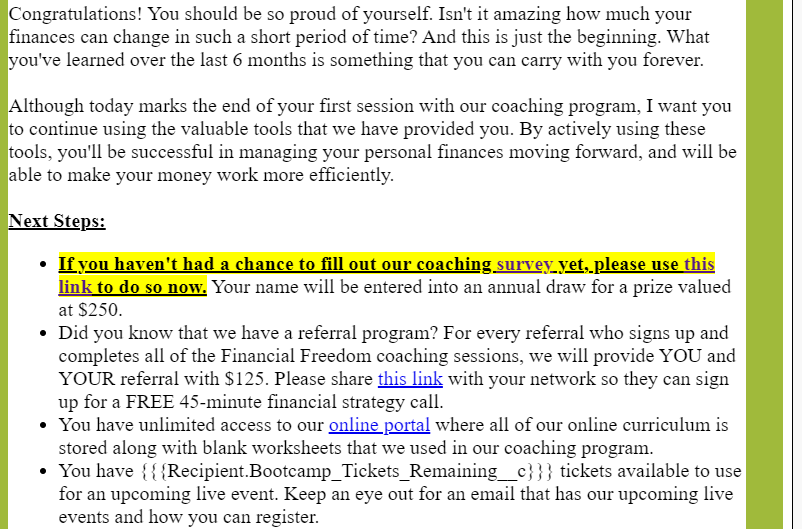


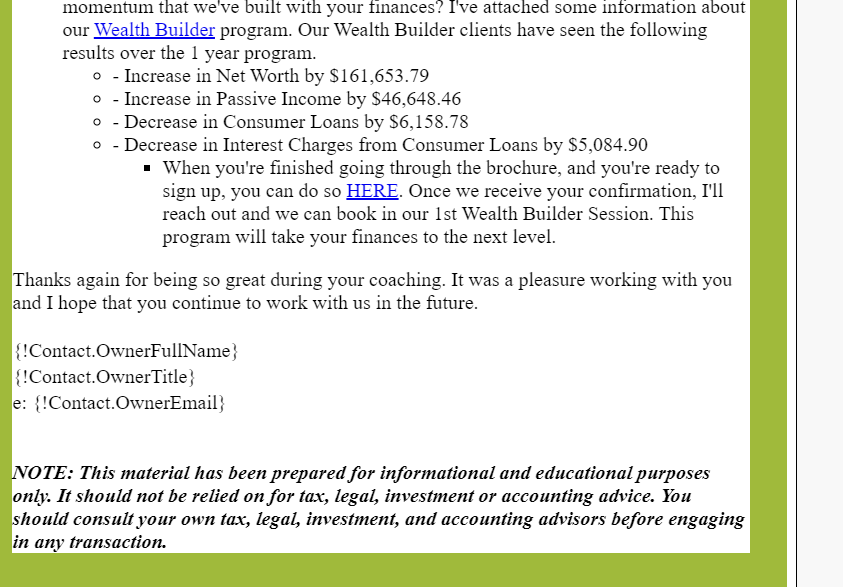
**Fig. 4.4 Process Builder**

**Fig. 4.5 CLASSIC EMAIL TEMPLATES**

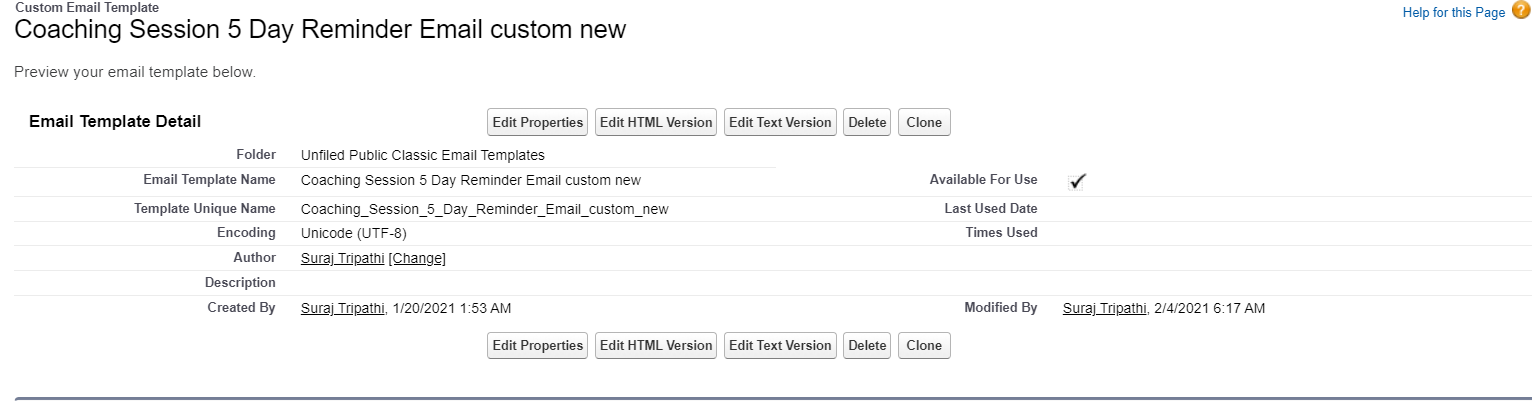
In Salesforce, you can create four different types of email templates: text, **HTML with Classic Letterhead**, custom, and Visualforce. All of these email templates can include text, merge fields, and attached files.

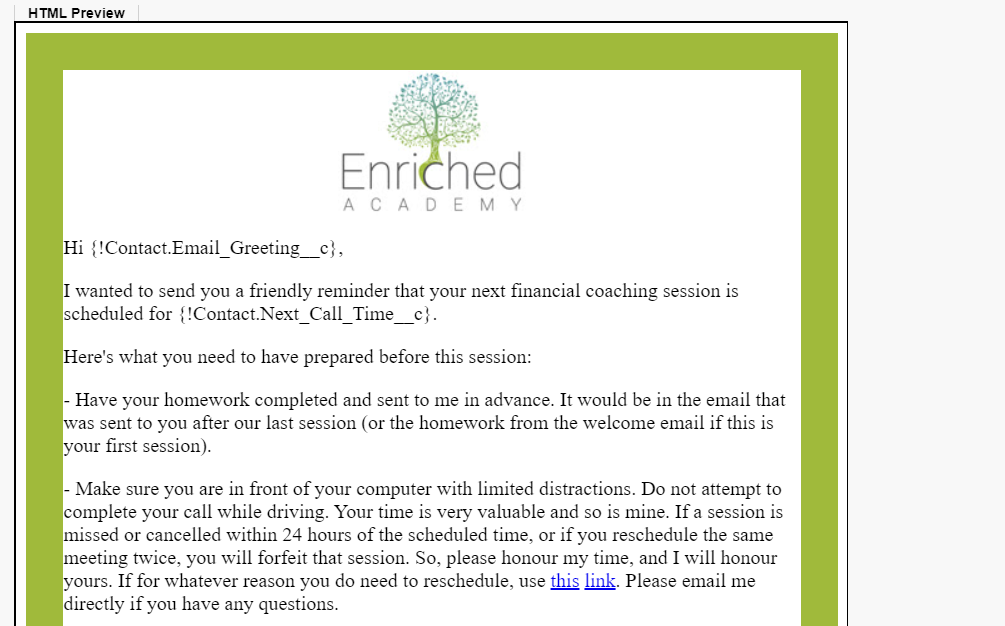


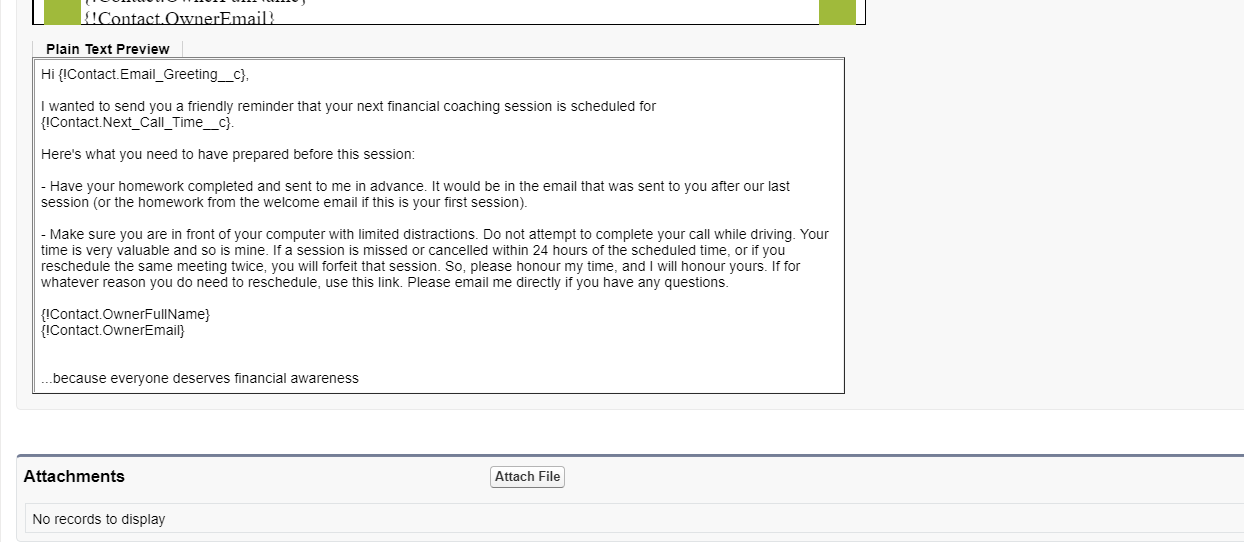




**Fig. 4.5.1 Classic Email Templates**

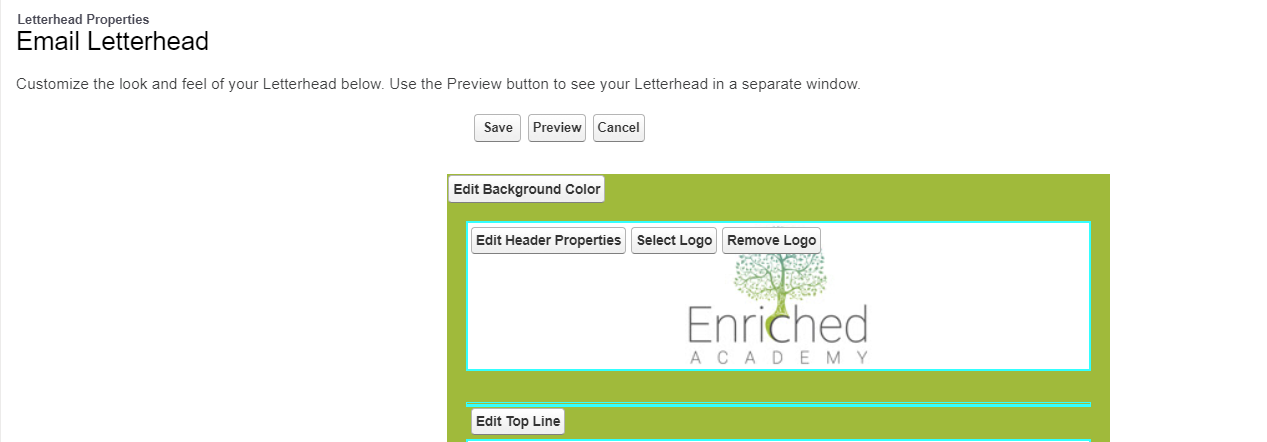


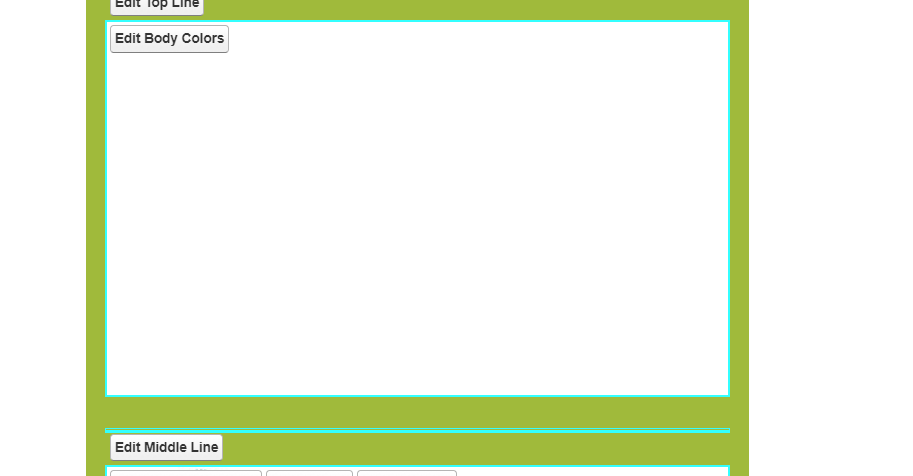


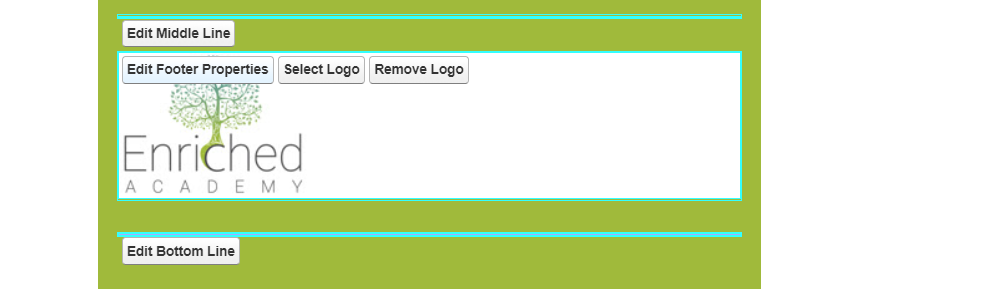


**Fig. 4.5.2 Classic Email Templates**

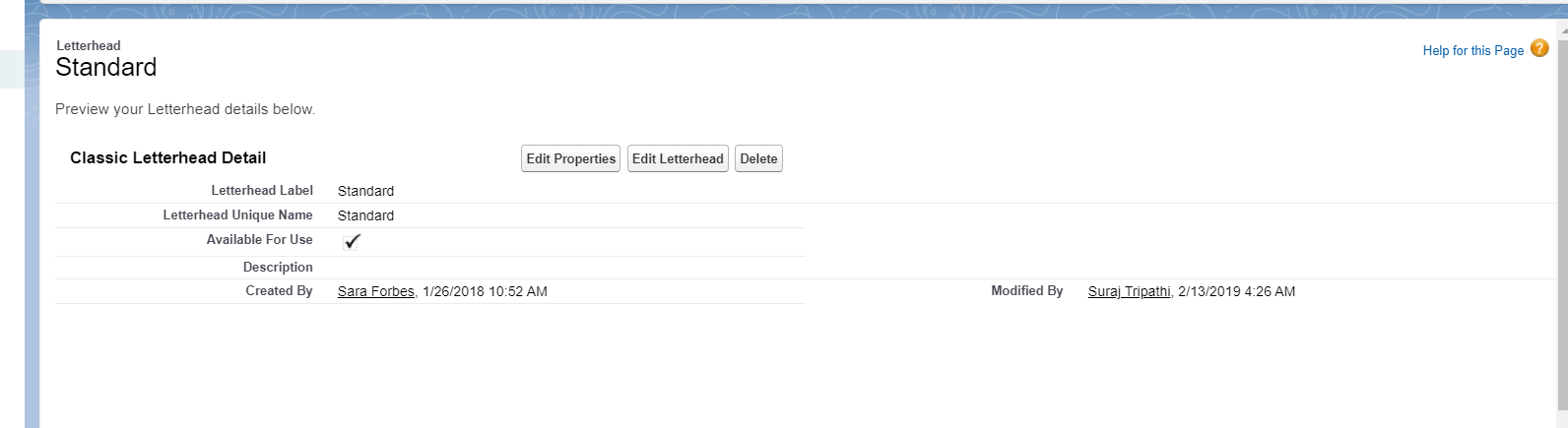
**Fig. 4.6 CLASSIC LETTERHEADS**

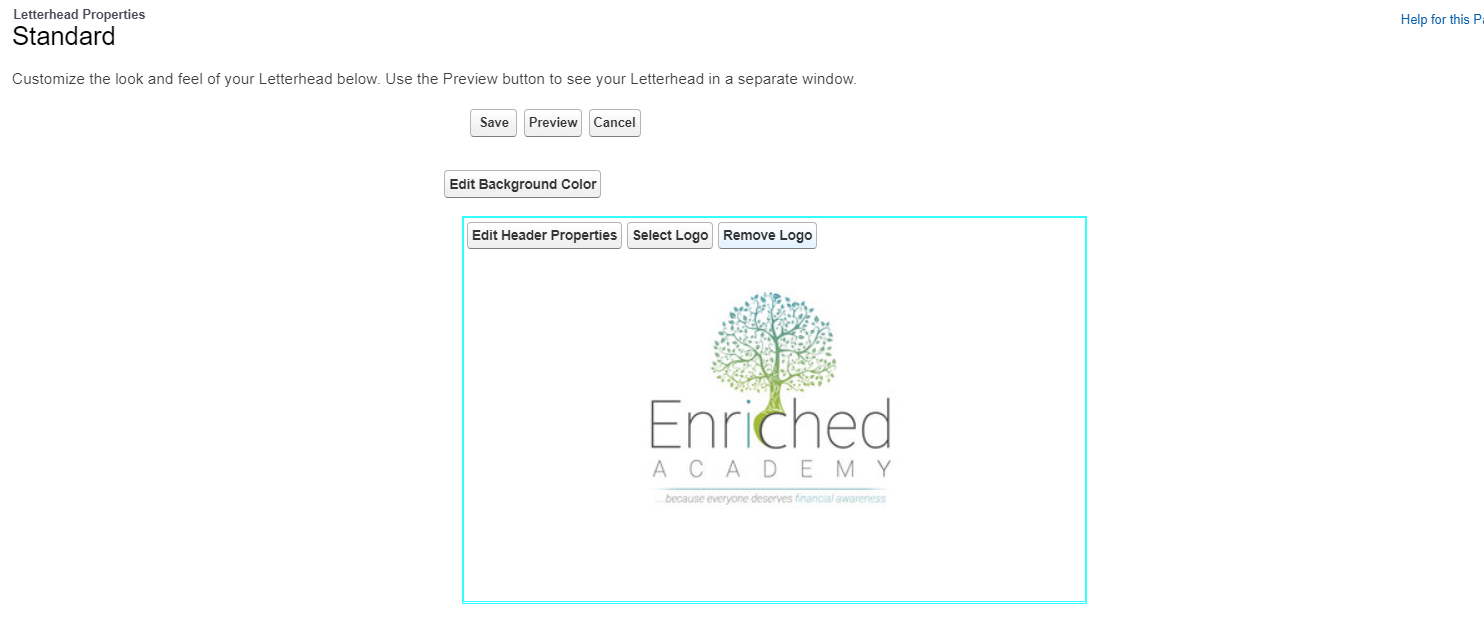






**Fig. 4.6.1 Classic Letterheads**







**Fig. 4.6.2 Standard Letterheads**

**CHAPTER 5**

**Extend Salesforce with Clicks, Not Code**

Ready to go beyond the basics of Salesforce administration? Want to customize your org, push its boundaries, and enhance its functionality? You can do that and so much more without writing a single line of code. All you need is your mouse and a sense of adventure. Enhance your objects, data, and fields, customize your org’s look and feel, augment your business processes, create websites, and even create apps—all using point-and-click tools.

The Force.com platform provides an enormous amount of functionality and flexibility, all of which is driven by underlying metadata. Force.com metadata is a collection of attributes that describe most components of data and applications that operate on the platform. Metadata describes the data structures in your environment, the declarative functionality implemented on the platform—even the applications you build on the platform.

To understand the power and reach of metadata, simply access a data record held in Force.com. When you access the record in your Force.com environment, the platform uses metadata to understand the structure of the record, the user interfaces defined for the record, the applications that use the user interface, even the security on that data and application.

**5.1 App Setup Menu**

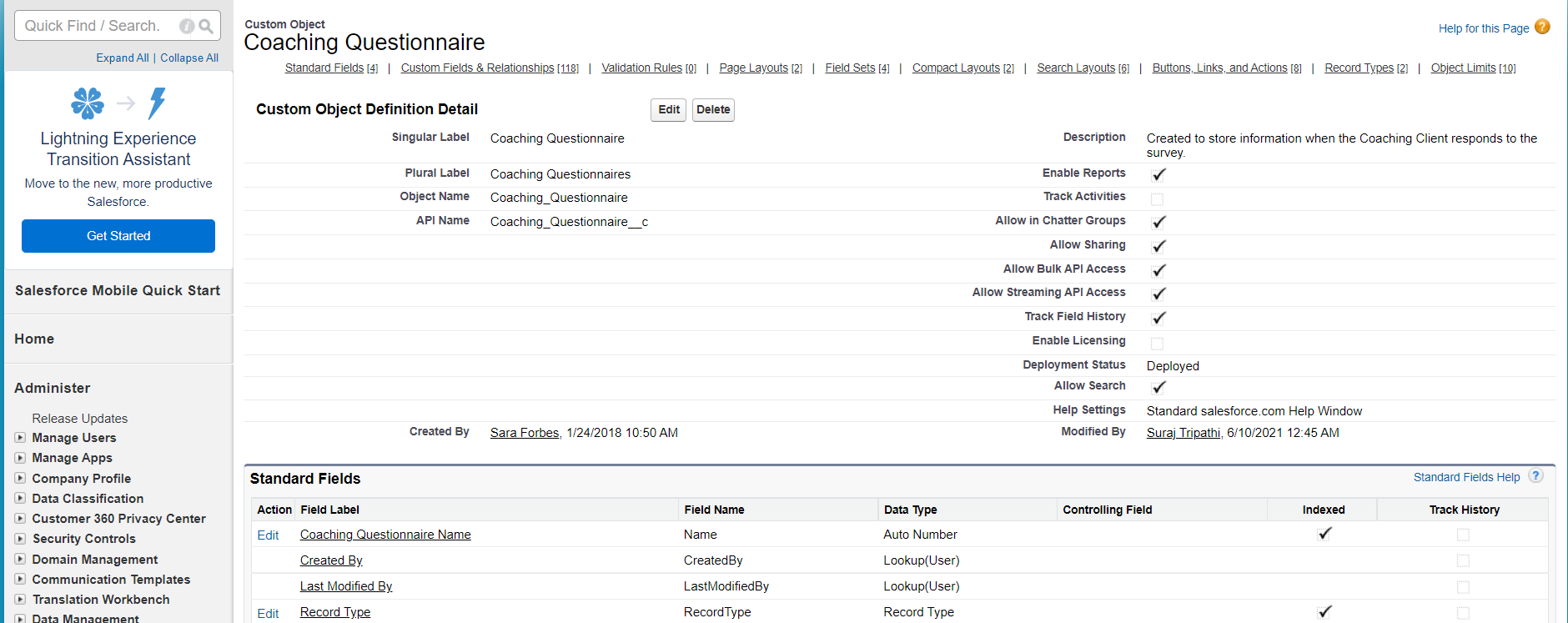
The App Setup menu is where you will spend the most time as a developer. These menus provide access to pages that let you create and configure Force.com components and services.

Once you create a custom object, you can edit the definition of the object via the Custom Object page.

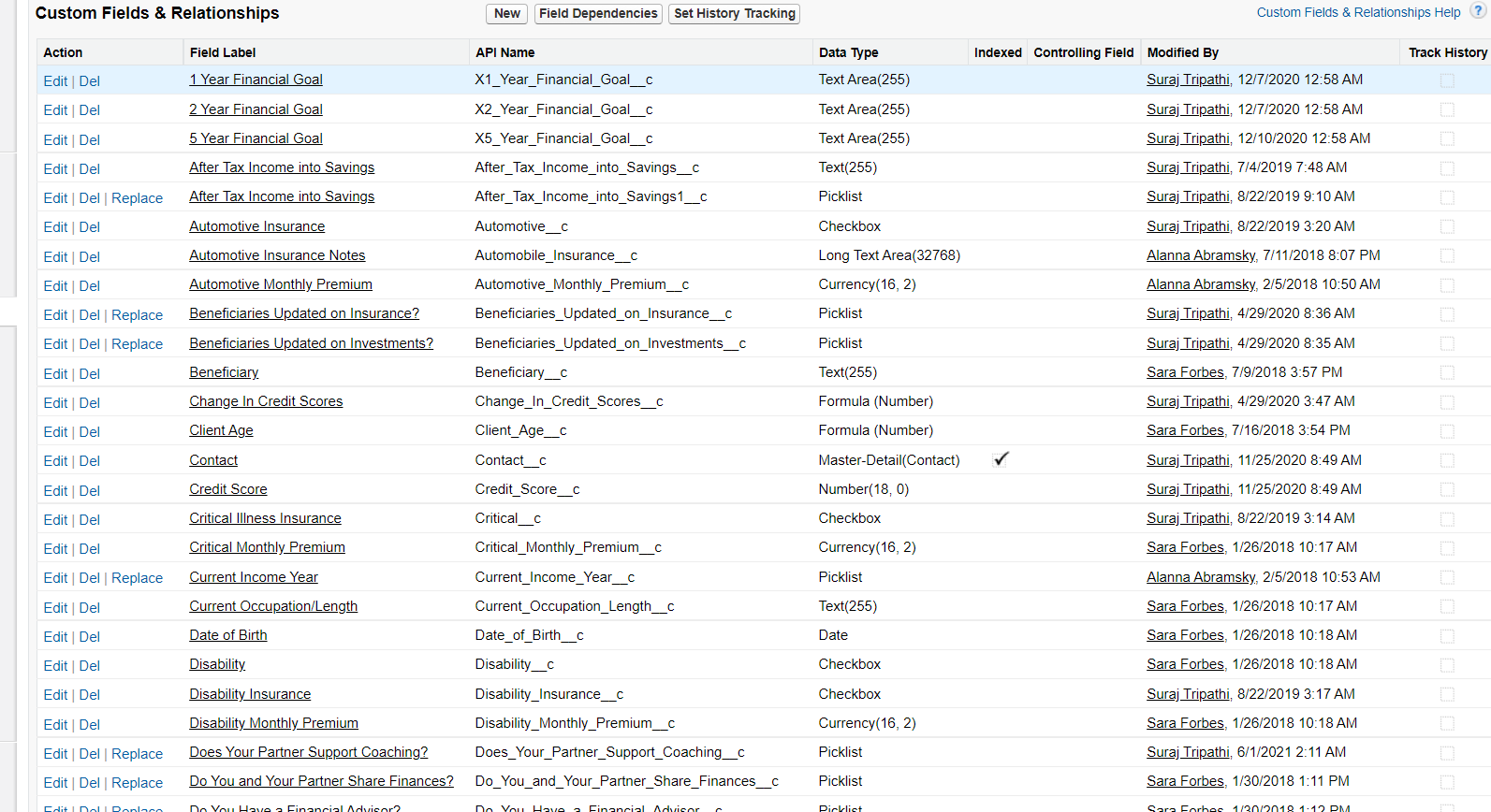
The Custom Object page provides links for adding custom fields, validations to enforce data integrity rules, database triggers, and custom buttons or links to the object's page layouts. You can also modify the attributes of standard fields, buttons, links or layouts for both the page and search dialogs, as well as add new page layouts or assign record types.

As an example, when you add a new field to a custom object, a wizard walks you through a number of steps, including:

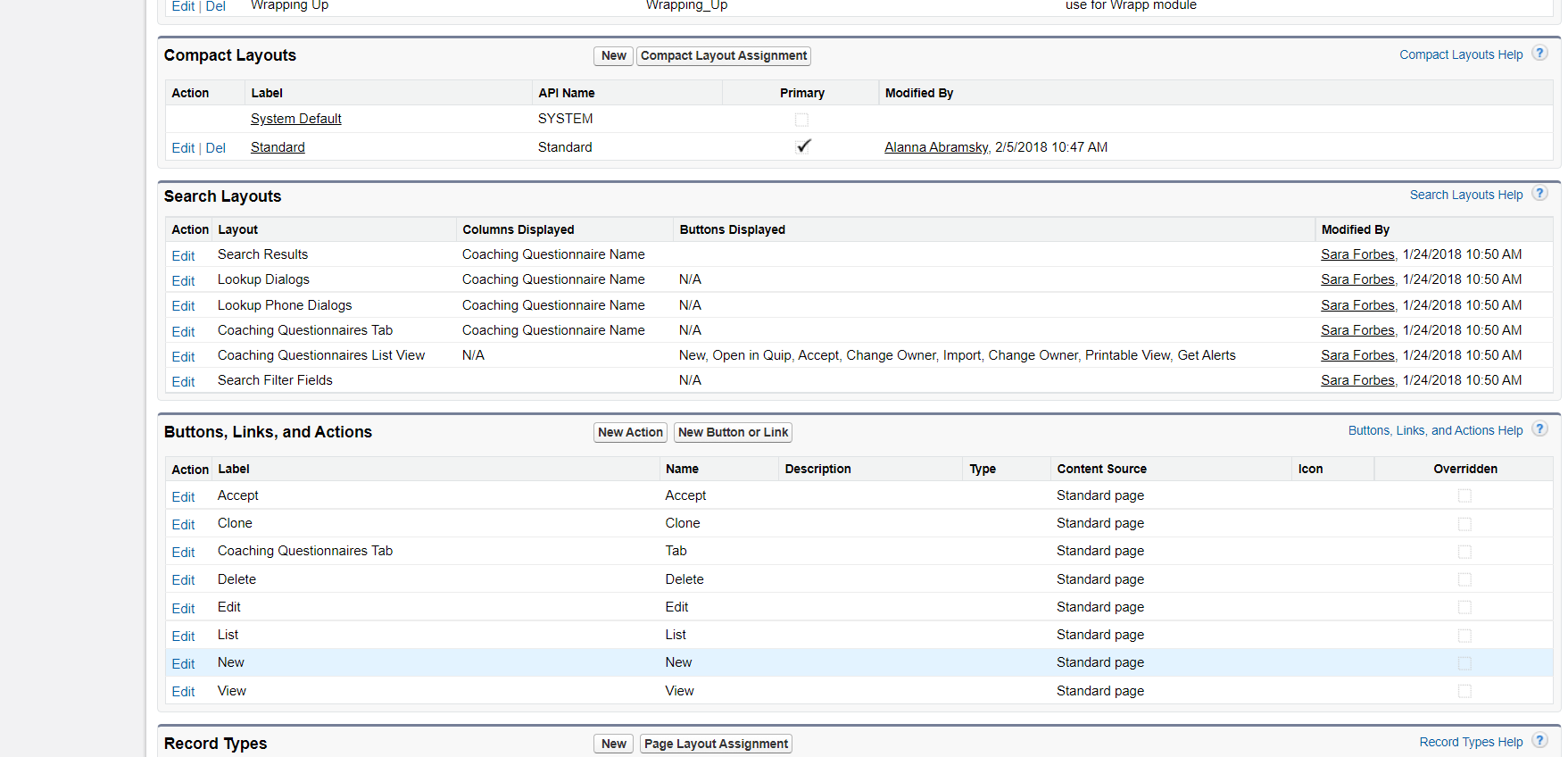
* Selecting a field type.
* Giving the field a label, name, help text, a default value, and potentially other attributes, such as the length of the field or whether a value is required.
* Assigning security settings to the field.
* Adding the field to existing page layouts.
* Depending on the type of field in focus, the wizard may include other pages for other relevant metadata attributes.



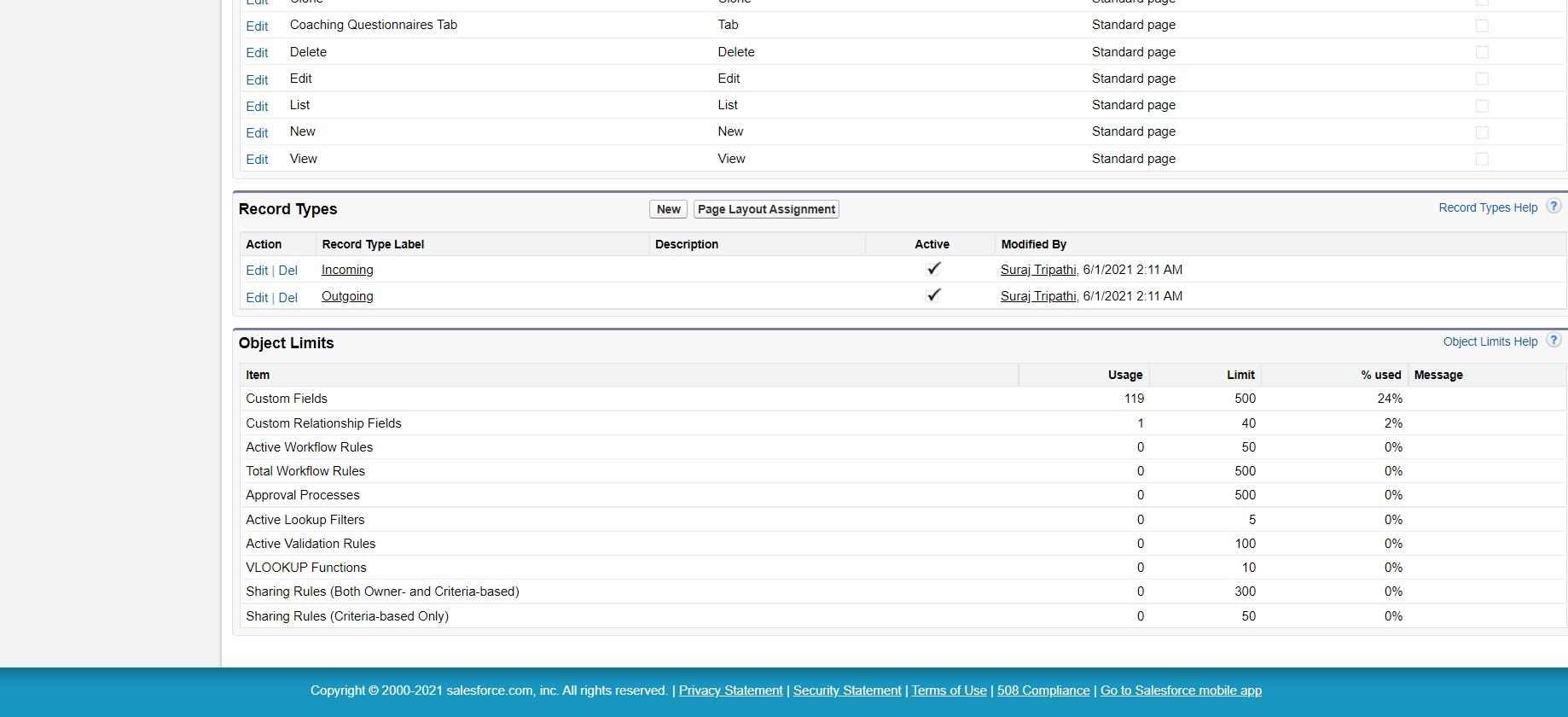
**Fig 5.1**



**Fig 5.2**



**Fig 5.3**



**Fig 5.4**

**CHAPTER 6**

**TESTING**

**SalesForce** **testing** requires the use of complex **test** methodologies as most of the features in **SalesForce** are built-in features that are customizable. When an issue is observed, the tester needs to make sure that he is **testing** the code that has been customized rather than **testing** the built-in **salesforce** functionality

SalesForce is built on a platform development language named as APEX. The language provides built-in unit test cases for developers to test their own code. The standard rule of SalesForce requires a developer to achieve 75% of code coverage with unit test cases.

From a tester's perspective, we should always aim for 100% code coverage within each test cycle.

#### **Salesforce Testing Process**

The salesforce testing process would be the same as that of testing a normal web-based application. However, a tester needs to have a clear perspective of the customizable features that are being built so that during the testing process, a tester can focus on those features alone rather than the built-in Salesforce features.

Testing of Salesforce applications requires a production like an environment called **SandBox.** Developers and testers need to use the Sandbox environment for each of their purposes.

Once the code is deployed in a Sandbox environment and approved to be ready for release, the code would be moved into production from the Sandbox environment. It is assumed that the tester has the basic knowledge of all the terms used in Salesforce before proceeding with testing.

#### **Salesforce Testing Tips**

**Salesforce testing must include the following features:**

* Testing must include UI testing, functional testing, regression testing, integration testing, system testing and system integration testing.
* Automation testing can also be enforced on SalesForce using tools like HP Unified Functional Testing (UFT ) and Selenium.
* A tester needs to be cautious during UI testing as most of the web pages on the Salesforce platform are Visual Force pages. The dynamic nature of visual force pages need to be paid special attention as all the elements of a webpage may not be loaded at one go.
* Testers need to create functional flows including positive and negative flows to cover the entire functionality of an application.
* Workflows using various user roles must be constructed and tested.
* Test cases need to be documented using a test management tool like HP ALM.
* Test Data needs to be prepared for validating the reports functionality.

#### **Roles and Responsibilities of a Salesforce Tester**

Testers involved in SalesForce are often referred to as **‘Quality Engineers’** as opposed to **‘Quality Assurance’** people as  SalesForce testing requires the testers to build complex test frameworks, understanding the functionality of an application in depth and the ability to work with the developers and project stakeholders.

Please note that some of the default functionalities provided by SalesForce cannot be removed, although your organization may not use them. Testers need to ignore the default functionality and focus on the customized functionality built by the organization.

**Given below are some of the major roles and responsibilities of a Salesforce tester.**

* A tester needs to have [clear communication](https://www.softwaretestinghelp.com/tester-and-developer-communication/) with the development team, to understand the customizable features that are being built into SalesForce.
* The tester needs to coordinate with the business whenever required as the requirement document for Salesforce is usually complex to understand and requires a lot of effort to be understandable by the testers.
* The tester needs to make sure that the code coverage does not go below 75% as per the standard Salesforce rule.
* The tester needs to conduct role-based testing to ensure the consistency of data with various user roles.
* The tester needs to perform compatibility testing of the third-party applications integrated with SalesForce if any.
* A tester needs to be familiar with load testing tools such as JMeter to validate the complex flows that produce inconsistent results in Salesforce.
* A tester needs to be familiar with multiple application flows.

#### **6.1 SalesForce Exploratory Testing**

**Exploratory Testing in Salesforce would involve the following best practices:**

* Testing should involve validating the consistency of data across multiple screens.
* UI Testing must involve documented test cases as per the requirement document.
* Testing should involve negative test flows, such as deleting the default data generated and validating the behaviour of an application.
* Testing should involve user input validation on the form fields.
* [Cross browser compatibility testing](https://www.softwaretestinghelp.com/how-is-cross-browser-testing-performed/) needs to be performed to ensure if the rendering of data is correct across multiple browsers.
* Testing must include Maximum length validation for each of the editable input fields along with the invalid data validation.
* Testing must also include error message validation when invalid data is passed onto the applications.
* Amount field validation on banking applications using Boundary Value Analysis technique needs to be performed with proper diligence.
* Reports and dashboard testing need to be paid special attention to various test data parameters.
* Testing should include the entire application flow, along with individual functional flows.
* Multiple permutations and combinations of functional flows can be tested for positive and [negative testing](https://www.softwaretestinghelp.com/what-is-negative-testing/).
* API testing for integrated third-party applications needs to be performed.
* Identify the default Salesforce functionalities that come in the way of customized features and coordinate with the developers.

#### **6.2 SalesForce Test Automation**

Automated functional testing of SalesForce is a challenging one as most of the web pages are dynamic in nature on the SalesForce platform. Hence, SalesForce demands automation testers to build robust automation framework to sustain in the future. Also, there can be frequent updates to the applications as they are on cloud applications.

**Test Automation on Salesforce can be achieved using any of the following tools:**

* Selenium web driver
* HP Unified Functional Testing (UFT)
* Test Frameworks, such as Cucumber
* Provar

#### **6.3 Salesforce Load Testing**

Load testing involves testing the behavior of an application under varying loads. SalesForce.com is a highly scalable platform built for handling a large number of users. Salesforce.com is tested by the platform developers themselves for performance bottlenecks.

However, load testing becomes essential when a newly introduced piece of code yields performance bottlenecks that have to be addressed. Load Testing on Salesforce platform can be performed using performance testing tools such as HP LoadRunner and Apache JMeter.

#### **6.4 Salesforce Security Testing**

Security testing on the Salesforce platform is usually done by SalesForce development team. Before placing a request for a security test, it is best to review the ‘Application and Network Vulnerability Assessment Summaries’ provided by Salesforce.

After reviewing the summary, if a security test is still required, then a Security Assessment Test can be scheduled with the Salesforce team.

**CHAPTER 7**

**REFERENCES**

The following references have been used by me, during all the phases of the project:

1. Krutarth Soni (2017), “Roadmap to salesforce security governance & salesforce access management”
2. Radhika Gupta (2018), “Custom Application Development in Cloud Environment: Using Salesforce”
3. Aneta Poniszewska-Maranda (2017), “Use of Salesforce Platform for Building Real-Time Service Systems in Cloud”
4. L.N. Goslin (1997), “New product development: salesforce and cycle time”
5. Steve Fisher (2007), “The Architecture of the Apex Platform, salesforce.com's Platform for Building On-Demand Applications”
6. Jigar Patel (2016), “An approach to introduce basics of Salesforce.com: A cloud service provider”
7. Amitabh Saxena (2013), “Detecting SOQL-injection vulnerabilities in SalesForce applications”
8. Jigar Patel (2017), “An integration of salesforce.com with Twitter: A case of AppExchange”
9. Kush R. Varshney (2013),  “Dose-response signal estimation and optimization for salesforce management”
10. S. Fisher (2006),  “Service Computing: The AppExchange Platform”
11. Alexandre Chaves Da Silva (2015), “A Case Study Using Testing Technique for Software as a Service (SaaS)”
12. Anuradha S. Kanade (2013), “Choosing right database system: Row or column-store”
13. Ming-Chien Shan (2010), “Let's Walk Out of the Cloud”
14. Sohan Singh Yadav (2010) “CLOUD: A computing infrastructure on demand”
15. Kasiviswanadhan, S (1998), “A practical approach to library automation”, Automation  in Libraries and Information Retrieval Units: RlLISAR Bulietin, vol.4, no.3